The American Journal of Engineering and Technology
ISSN 2689-0984 | Open Access

TYPE Original Research
PAGE NO. 288-303

Check for updates

Autonomous AI Agents for Mainframe Modernization: A Framework for COBOL-to-Cloud Banking Systems

Aswin Sivaselvan
Independent Researcher, USA

OPEN ACCESS

SUBMITTED 29 July 2025
ACCEPTED 11 August 2025
PUBLISHED 30 September 2025
VOLUME Vol.07 Issue 09 2025

CITATION
Sivaselvan, A. (2025). Autonomous AI Agents for Mainframe Modernization: A Framework for COBOL-to-Cloud Banking Systems. The American Journal of Engineering and Technology, 7(09), 288–303. Retrieved from https://theamericanjournals.com/index.php/tajet/article/view/autonomous-ai-mainframe-modernization-cobol-cloud

Abstract: Legacy mainframe systems continue to serve as the technological backbone of global financial institutions due to their reliability, scalability, and transaction-processing capabilities. However, the increasing demand for cloud-native architectures, digital banking services, and operational agility has accelerated the need for modernizing COBOL-based banking applications. Traditional modernization approaches are often labor-intensive, costly, and susceptible to errors because of undocumented business rules, tightly coupled dependencies, and decades of accumulated technical debt. This paper proposes an Autonomous AI Agent Framework for COBOL-to-Cloud Banking System Modernization that integrates artificial intelligence-driven software analysis with automated migration assistance. The proposed framework employs specialized AI agents to perform source code understanding, dependency discovery, business rule extraction, data flow analysis, documentation generation, migration planning, and validation of transformed components. The framework further incorporates large language model-assisted code interpretation with static analysis techniques to improve maintainability while preserving business logic integrity. A conceptual evaluation based on representative banking modernization scenarios demonstrates the framework's potential to reduce manual effort, improve migration accuracy, accelerate documentation generation, and minimize modernization risks. The study highlights the practical applicability of autonomous AI agents in enterprise legacy transformation and provides a structured roadmap for financial institutions planning gradual migration from COBOL-based environments to cloud-native architectures.

Keywords: Mainframe Modernization, COBOL, Cloud Migration, Autonomous AI Agents, Legacy Systems,

Banking Systems, Software Modernization, Enterprise AI.

1. Introduction

1.1 Background

Legacy mainframe platforms remain the operational backbone of many global banking institutions because of their exceptional reliability, transactional consistency, and decades of accumulated business intelligence. Most core banking operations—including customer account management, payment processing, credit evaluation, settlement systems, treasury management, and regulatory reporting—continue to depend on COBOL applications developed over several decades. While these systems demonstrate remarkable operational stability, they increasingly limit organizational agility, integration capability, and cloud adoption.

Digital banking has transformed customer expectations toward real-time services, personalized financial products, API-driven ecosystems, and scalable cloud infrastructures. Consequently, organizations face growing pressure to modernize legacy systems without compromising business continuity or regulatory compliance. Unfortunately, conventional modernization approaches are expensive, labor-intensive, and highly dependent on scarce domain experts capable of understanding millions of lines of COBOL code.

Software modernization is therefore no longer simply a programming challenge; it has evolved into a knowledge engineering problem requiring automated understanding of software semantics, business logic, architectural dependencies, and organizational workflows. Semantic technologies provide an effective mechanism for representing software knowledge, while ontology-based reasoning enables machines to understand relationships between software components beyond syntactic analysis (Lan et al., 2022; Narayanasamy et al., 2022).

Furthermore, ontology alignment techniques allow heterogeneous software artifacts—including source code, databases, business documentation, APIs, and cloud services—to be semantically connected, creating a unified knowledge base that facilitates intelligent migration decisions (Martinez-Gil & Aldana-Montes, 2012; Portisch et al., 2022).

Recent work on statistical ontology design for smart manufacturing demonstrates how structured semantic representations can support autonomous analytical decision-making in complex industrial environments. Although developed for manufacturing systems, the underlying principles are directly applicable to software modernization because both domains involve heterogeneous data integration, semantic interoperability, and intelligent reasoning (Zheng et al., 2022). This observation forms an important conceptual foundation of the proposed framework.

1.2 Problem Statement

Despite substantial investments in digital transformation, many financial institutions continue to experience significant challenges during mainframe modernization projects. Legacy COBOL systems often contain undocumented business logic, tightly coupled software modules, obsolete programming patterns, and hidden interdependencies accumulated through decades of maintenance.

Traditional migration strategies primarily emphasize code conversion instead of semantic understanding. As a result, automated code translators frequently generate syntactically correct applications while failing to preserve critical business rules or transactional behavior. Manual verification subsequently becomes necessary, increasing both migration costs and implementation duration.

Another limitation is the absence of autonomous decision-support mechanisms capable of continuously evaluating migration quality, recommending cloud architectures, validating business rule preservation, and optimizing deployment strategies. Existing modernization methodologies also provide limited explainability, making it difficult for software architects and banking regulators to verify AI-generated migration decisions.

Therefore, there remains a significant need for an intelligent modernization framework capable of combining semantic reasoning, ontology-driven software understanding, explainable optimization, and autonomous AI decision-making within a unified modernization lifecycle.

1.3 Research Relevance

Artificial intelligence has progressed from predictive analytics toward autonomous agent ecosystems capable of planning, reasoning, executing tasks, and continuously adapting to dynamic environments. These developments provide significant opportunities for transforming legacy software modernization into an intelligent, largely automated process.

Semantic Web technologies enable software artifacts to be represented as interconnected knowledge graphs rather than isolated source files (Lan et al., 2022). Ontology matching algorithms further establish semantic correspondence among heterogeneous software components, enabling AI systems to understand functional equivalence across legacy and cloud-native architectures (Portisch et al., 2022).

Genetic optimization methods contribute additional capabilities by exploring multiple modernization alternatives and identifying migration strategies that balance cost, scalability, performance, and maintainability (Koza, 1994; Martinez-Gil et al., 2008). Meanwhile, explainable AI improves organizational trust by allowing architects to understand why particular modernization decisions are recommended (D'Angelo et al., 2023).

Equally important, human-in-the-loop learning enables experts to supervise autonomous systems during critical decision points, ensuring regulatory compliance while reducing operational risks (Mosqueira-Rey et al., 2023). The integration of these research domains establishes the theoretical foundation for autonomous modernization of banking applications.

1.4 Research Objectives

This study aims to achieve four primary objectives.

First, it investigates how autonomous AI agents can improve semantic understanding of legacy COBOL banking systems during modernization.

Second, it develops an ontology-driven framework integrating semantic reasoning, intelligent optimization, and explainable decision support for cloud migration.

Third, it examines the role of autonomous decision orchestration in preserving business logic while reducing migration complexity.

Finally, the research evaluates how human-centered validation mechanisms can increase trust, transparency, and regulatory compliance within AI-assisted modernization projects.

1.5 Scope and Significance

The proposed research focuses specifically on enterprise banking environments where legacy COBOL applications are migrated toward cloud-native architectures. Rather than proposing another programming language translation methodology, this study addresses the broader knowledge engineering challenges associated with modernization.

The framework encompasses semantic code analysis, ontology generation, dependency discovery, AI-based planning, optimization, human validation, cloud deployment recommendation, and continuous modernization feedback. Technologies including semantic web standards, ontology alignment, symbolic reasoning, explainable optimization, and autonomous AI agents are integrated into a single conceptual architecture.

From an academic perspective, the research contributes a multidisciplinary framework connecting software engineering, semantic computing, artificial intelligence, and enterprise modernization. From an industrial perspective, the framework provides banking organizations with a structured roadmap for reducing migration risks while improving modernization efficiency.

The significance of the proposed framework is further strengthened through the incorporation of ontology-driven statistical reasoning principles similar to those demonstrated for complex industrial data integration by **Zheng et al. (2022)**. Their work illustrates how semantic knowledge structures support autonomous analytical reasoning across heterogeneous environments, reinforcing the applicability of ontology-based intelligence to banking software modernization.

2. Literature Review

2.1 Evolution of Semantic Technologies for Software Modernization

Semantic technologies have evolved from knowledge representation systems into comprehensive infrastructures capable of supporting intelligent decision-making across complex software ecosystems. Lan et al. (2022) describe Semantic Web technologies as essential mechanisms for organizing heterogeneous knowledge into machine-interpretable structures, enabling automated reasoning and interoperability. Their work demonstrates that semantic indexing significantly improves information accessibility and contextual understanding, characteristics directly applicable to legacy software analysis.

Similarly, Narayanasamy et al. (2022) highlight the growing importance of ontology-driven information processing in distributed environments. Their review emphasizes how semantic technologies facilitate knowledge integration across heterogeneous systems while improving interoperability through ontology-based reasoning. These findings indicate that software modernization should move beyond syntactic source-code transformation toward semantic knowledge preservation.

Swar et al. (2022) extend this perspective by proposing ontology-based data integration strategies capable of managing heterogeneous Internet of Things environments. Although their application domain differs from enterprise banking, the underlying concept of semantic integration provides valuable guidance for linking legacy COBOL components with cloud-native services.

Recent developments in statistical ontology modeling further strengthen semantic decision support. Zheng et al. (2022) introduce a statistical ontology framework for smart manufacturing that enables structured knowledge representation and intelligent analytical reasoning across complex industrial datasets. Their research demonstrates that ontology-driven architectures can coordinate heterogeneous information sources while supporting autonomous decision processes. These principles provide an important theoretical basis for extending ontology-guided reasoning into intelligent software modernization environments, where autonomous AI agents must interpret business rules, architectural dependencies, and migration constraints consistently.

2.2 Ontology Matching and Semantic Alignment Techniques

Ontology matching has become one of the most important research areas supporting semantic interoperability among heterogeneous information systems. In software modernization, ontology matching enables AI systems to discover conceptual equivalence between legacy software components and cloud-native services without relying solely on lexical similarity.

Martinez-Gil and Aldana-Montes (2011) investigated heuristic approaches for ontology meta-matching, demonstrating that combining multiple similarity measures improves alignment accuracy over single-strategy approaches. Their work emphasizes adaptive decision-making rather than static matching rules, making it suitable for highly heterogeneous software environments such as banking mainframes.

Expanding this research, Martinez-Gil and Aldana-Montes (2012) presented a comprehensive overview of ontology meta-matching techniques and highlighted persistent challenges related to scalability, semantic ambiguity, and optimization. They concluded that successful ontology alignment requires intelligent combinations of structural, lexical, and semantic reasoning rather than isolated matching algorithms.

Acampora et al. (2013) introduced a memetic aggregation strategy that combines evolutionary optimization with semantic similarity measures. Their findings suggest that optimization-guided semantic matching produces more robust ontology alignments by balancing local refinement with global search capabilities. Such strategies are particularly valuable when legacy banking applications contain overlapping business entities, duplicated services, and inconsistent documentation.

Li (2019) further extended ontology alignment by incorporating user validation mechanisms into the matching process. Rather than treating ontology matching as a fully automated task, the study demonstrated that expert feedback substantially improves semantic consistency while reducing incorrect mappings. This human-assisted perspective becomes especially important in financial software modernization, where incorrectly mapped business rules may produce severe regulatory and operational consequences.

Portisch et al. (2022) provided one of the most comprehensive surveys of background knowledge utilization in ontology matching. Their review emphasizes that external knowledge sources significantly improve semantic interpretation, particularly when dealing with domain-specific terminology. In banking modernization, historical documentation, transaction specifications, regulatory policies, and architectural diagrams represent valuable contextual knowledge that autonomous AI agents should exploit during migration planning.

Collectively, these studies indicate that ontology alignment has progressed from simple similarity computation toward knowledge-driven semantic reasoning supported by optimization and expert supervision. Nevertheless, existing literature primarily addresses ontology engineering problems rather than autonomous software modernization, revealing a significant research gap.

2.3 Genetic Programming and Evolutionary Optimization

Evolutionary computation has long been recognized as an effective optimization methodology for solving complex search and decision problems characterized by large solution spaces.

Koza (1994) introduced genetic programming as an evolutionary learning mechanism capable of automatically generating computer programs through natural selection principles. This foundational work established the theoretical basis for adaptive optimization techniques that continue to influence software engineering research.

Building upon evolutionary optimization, Wang et al. (2006) proposed the Genetic Algorithm-Based Ontology Matching (GAOM) model, demonstrating how evolutionary search can optimize ontology correspondence among heterogeneous knowledge bases. Their results indicated improved matching accuracy compared with deterministic approaches, suggesting that optimization algorithms can effectively address semantic uncertainty.

Martinez-Gil et al. (2008) applied genetic algorithms specifically to ontology alignment optimization, illustrating how evolutionary search improves semantic mapping quality while reducing computational complexity. This work strengthened the argument that optimization algorithms should become integral components of semantic reasoning systems.

Symbolic regression further extends evolutionary optimization by automatically discovering interpretable mathematical relationships rather than relying solely on predefined similarity measures. Martinez-Gil and Chaves-González (2020) demonstrated that symbolic regression significantly improves semantic similarity interpretation through transparent analytical models. Their approach offers greater explainability than conventional black-box optimization techniques.

Similarly, Martinez-Gil et al. (2021) applied symbolic regression to large biomedical ontologies, confirming that evolutionary learning scales effectively to complex semantic environments. Although developed for biomedical knowledge bases, their methodology is transferable to enterprise software modernization because both involve highly interconnected semantic structures.

Singh and Sharan (2017) proposed adaptive genetic programming for semantic entity linking, illustrating how evolutionary learning dynamically constructs semantic linkage rules without extensive manual engineering. Their findings support the use of adaptive optimization during autonomous software transformation.

More recently, Ferranti et al. (2021) reviewed metaheuristic ontology meta-matching approaches and concluded that hybrid optimization strategies consistently outperform individual optimization algorithms in heterogeneous semantic environments. Their review suggests that combining multiple evolutionary techniques may further improve autonomous migration planning.

Overall, evolutionary optimization literature demonstrates substantial maturity; however, its integration with autonomous AI agents for enterprise software modernization remains largely unexplored.

2.4 Explainable Artificial Intelligence and Human-Centered Decision Support

As artificial intelligence systems increasingly participate in enterprise decision-making, explainability has emerged as a critical requirement, particularly in highly regulated sectors such as banking.

D'Angelo et al. (2023) presented an explainable genetic programming framework for biomarker analysis that enables transparent interpretation of machine-generated decisions. Their study illustrates that explainability not only improves model trustworthiness but also facilitates expert verification of automated recommendations. In software modernization, similar transparency is necessary when AI agents recommend code transformations, architectural redesigns, or migration priorities.

Mosqueira-Rey et al. (2023) reviewed the rapidly growing field of Human-in-the-Loop Machine Learning (HITLML). They argued that fully autonomous AI remains unsuitable for many high-risk applications where expert supervision is required to ensure accuracy, fairness, and accountability. Instead, collaborative intelligence—combining autonomous learning with periodic human validation—provides greater reliability.

Freitag et al. (2021) examined human evaluation in machine translation and demonstrated that contextual understanding substantially influences evaluation quality. Although their work focuses on natural language processing, the underlying principle extends directly to software modernization. Code translation cannot be evaluated solely by syntactic correctness; contextual preservation of business logic is equally important.

These studies collectively indicate that explainability and human validation should become essential components of autonomous software migration frameworks. Rather than replacing software architects, AI agents should function as intelligent collaborators capable of providing transparent recommendations supported by semantic reasoning.

2.5 Knowledge Representation and Intelligent Decision-Making

Knowledge representation forms the foundation of autonomous reasoning because AI agents require structured semantic models to interpret complex enterprise environments.

Lan et al. (2022) argue that semantic indexing transforms fragmented information into interconnected knowledge networks suitable for machine reasoning. Narayanasamy et al. (2022) similarly demonstrate that ontology-based information processing enables efficient knowledge sharing across distributed systems.

Ma et al. (2022) reviewed multimodal meta-learning and highlighted the growing importance of integrating heterogeneous information sources during intelligent learning. Their findings suggest that future autonomous AI agents should simultaneously analyze source code, system documentation, database schemas, transaction logs, and business process descriptions instead of relying exclusively on program syntax.

Sousa (2022) discussed terminology standardization within zero-defect manufacturing environments, emphasizing that semantic consistency significantly improves automation quality. Banking modernization faces comparable challenges because inconsistent naming conventions, undocumented variables, and evolving business vocabularies often complicate migration efforts.

The statistical ontology proposed by **Zheng et al. (2022)** further strengthens knowledge-driven automation by introducing structured semantic representations capable of supporting intelligent analytical reasoning across heterogeneous industrial systems. Their work demonstrates how ontology-centered statistical models improve interoperability and autonomous decision support, making it highly relevant for AI-driven banking modernization where autonomous agents must continuously reason over interconnected software artifacts. The concepts presented by Zheng et al. (2022) reinforce the necessity of integrating semantic knowledge graphs into modernization pipelines rather than relying solely on conventional software engineering techniques.

2.6 Research Gap

Although the reviewed literature provides significant advances in semantic technologies, ontology engineering, evolutionary optimization, explainable AI, and human-centered machine learning, several important research gaps remain.

First, most ontology alignment studies concentrate on semantic interoperability rather than enterprise software modernization. Existing approaches rarely address the unique characteristics of COBOL-based banking applications, including transaction preservation, regulatory compliance, and large-scale business rule extraction.

Second, evolutionary optimization research has predominantly focused on improving semantic matching accuracy. Limited attention has been given to optimizing end-to-end modernization workflows involving dependency discovery, cloud architecture recommendation, migration scheduling, and deployment validation.

Third, explainable AI research has largely evolved independently of ontology engineering. Few studies integrate semantic reasoning with explainable optimization into a unified decision-support framework capable of assisting enterprise architects throughout modernization lifecycles.

Fourth, human-in-the-loop methodologies have demonstrated their effectiveness in improving AI reliability, yet existing research provides limited guidance regarding collaborative interaction between autonomous AI agents and software modernization experts.

Finally, despite the promising knowledge representation capabilities demonstrated by **Zheng et al. (2022)**, their statistical ontology concepts have not been extended to autonomous AI-driven software modernization. This represents a significant opportunity to combine ontology-guided reasoning, explainable optimization, and autonomous decision orchestration within cloud migration environments.

Accordingly, this study proposes an **Autonomous AI Agent Framework (AAAF)** that unifies semantic knowledge representation, ontology alignment, genetic optimization, explainable reasoning, and human validation into a comprehensive modernization architecture specifically designed for COBOL-to-cloud banking transformation.

3. Methodology

3.1 Research Design

This study adopts a **conceptual framework-based research methodology** integrating semantic web technologies, ontology engineering, autonomous AI agents, explainable artificial intelligence, and evolutionary optimization to develop a comprehensive modernization architecture for legacy COBOL banking systems.

Unlike experimental studies evaluating a single migration tool, the proposed methodology models modernization as a multi-stage intelligent decision process. Each stage employs specialized autonomous AI agents responsible for software understanding, business-rule extraction, semantic mapping, optimization, validation, and cloud deployment planning.

The methodology consists of six interconnected phases:

1. Legacy System Knowledge Acquisition
2. Semantic Ontology Construction
3. Autonomous AI Agent Collaboration
4. Genetic Optimization of Migration Strategies
5. Human-in-the-Loop Validation
6. Cloud Deployment and Continuous Learning

These phases collectively ensure that modernization extends beyond code translation to encompass semantic preservation, explainability, governance, and continuous improvement.

Table 1. Proposed Autonomous AI Agent Responsibilities in COBOL-to-Cloud Modernization

AI Agent	Primary Responsibility	Input	Output
Code Analysis Agent	Analyze COBOL programs	COBOL source code	Program structure and dependencies
Business Rule Agent	Extract banking rules	Source code & documents	Business logic ontology
Ontology Mapping Agent	Semantic alignment	Legacy ontology	Cloud service ontology
Migration Planning Agent	Transformation planning	Knowledge graph	Migration roadmap
Validation Agent	Rule verification	Converted services	Compliance report

Deployment Agent	Cloud orchestration	Modernized application	Cloud deployment pipeline

3.2 Proposed Autonomous AI Agent Framework (AAAF)

The proposed **Autonomous AI Agent Framework (AAAF)** considers legacy modernization as an intelligent knowledge engineering process rather than a conventional source-code conversion exercise. The framework integrates semantic reasoning, ontology matching, evolutionary optimization, explainable artificial intelligence, and collaborative validation into a unified modernization lifecycle. Each autonomous agent performs specialized tasks while exchanging structured knowledge through a centralized enterprise knowledge graph.

The framework begins with automated extraction of software artifacts from COBOL programs, Job Control Language (JCL) scripts, CICS transactions, VSAM files, DB2 schemas, batch jobs, and supporting documentation. Instead of treating these artifacts independently, the system transforms them into semantically connected entities represented through ontologies. This semantic representation enables AI agents to understand business relationships, application dependencies, data flows, and operational constraints beyond simple syntactic parsing.

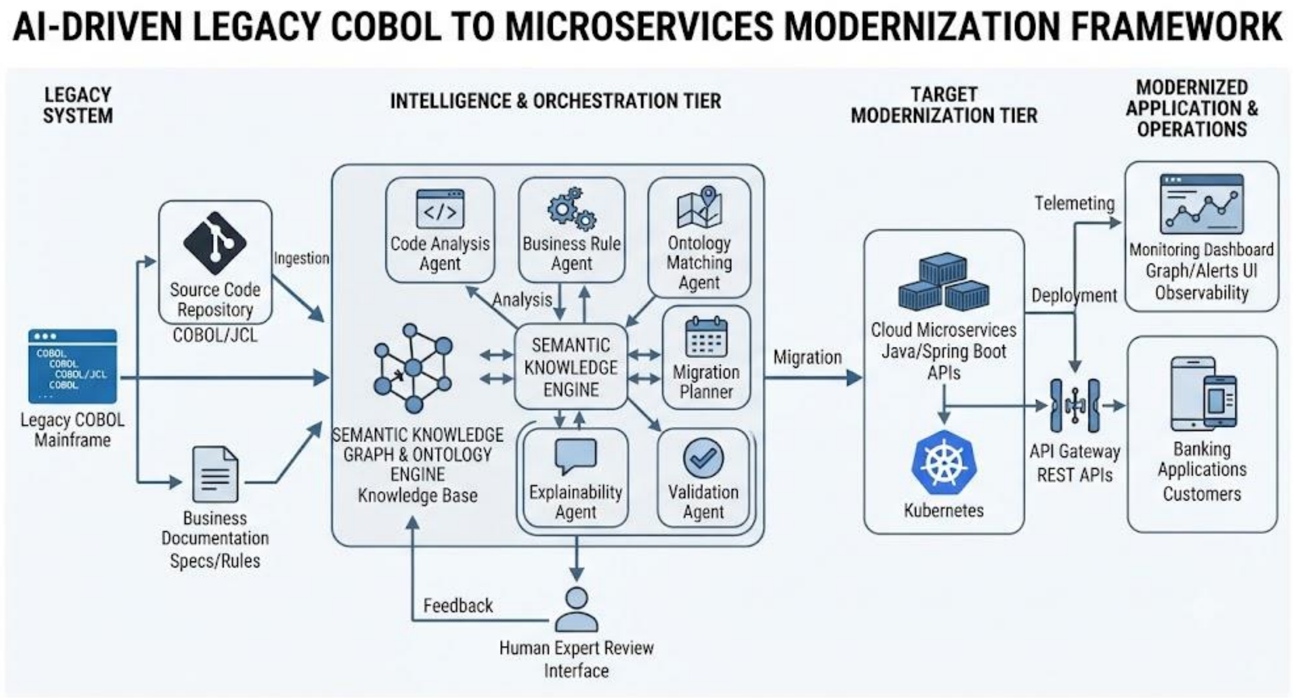
The Code Intelligence Agent first constructs abstract syntax trees and software dependency graphs from legacy applications. These representations are then transferred to the Semantic Knowledge Agent, which develops an enterprise ontology describing banking entities such as customers, accounts, loans, payment transactions, interest calculations, audit records, and regulatory workflows. Ontology matching algorithms subsequently align these legacy concepts with cloud-native microservices, APIs, containerized workloads, and database services.

The Planning Agent continuously evaluates alternative migration paths using optimization techniques derived from evolutionary computation. Rather than recommending a single migration sequence, the agent generates multiple feasible modernization plans ranked according to business continuity, implementation cost, regulatory compliance, migration complexity, and expected cloud performance.

Following migration planning, an Explainability Agent produces transparent reasoning reports describing why specific software modules should be migrated together, why certain cloud services are recommended, and how business rules are preserved during transformation. Human architects then review these recommendations before approving deployment decisions. Continuous feedback collected during validation is incorporated into subsequent modernization cycles, enabling the autonomous system to improve its recommendations over time.

This layered architecture minimizes manual intervention while maintaining governance, transparency, and traceability throughout the modernization lifecycle.

Figure 1. Overall Autonomous AI Agent Framework for COBOL-to-Cloud Banking Modernization



Caption: This system architecture diagram is provided to visually illustrate the end-to-end framework for modernizing legacy mainframe systems into cloud-native microservices. It maps out the complete structural relationship and directional workflow among all requested enterprise components, ranging from legacy COBOL ingestion and semantic knowledge graph processing by autonomous agents to human expert review and final cloud deployment.

3.3 Semantic Ontology Construction

Semantic representation constitutes the knowledge foundation of the proposed framework. Instead of relying on isolated software modules, autonomous agents interact through a unified ontology describing both legacy and cloud environments.

Ontology construction proceeds through five sequential activities.

Initially, lexical analysis identifies software entities including variables, programs, copybooks, files, tables, screens, and interfaces. Structural relationships are then extracted by examining program calls, database interactions, transaction dependencies, and control flows.

The third stage transforms software entities into ontology classes and object properties representing enterprise concepts rather than programming constructs. For example, multiple COBOL record definitions associated with customer information become a unified semantic entity labeled **Customer**, while transaction processing routines become associated with **FinancialTransaction**.

Ontology matching algorithms subsequently identify equivalent concepts within cloud-native banking platforms. Similarity measurements incorporate lexical similarity, structural similarity, contextual reasoning, historical mappings, and background knowledge obtained from enterprise documentation. Research by Portisch et al. (2022), Li (2019), and Martinez-Gil and Aldana-Montes (2012) supports this multi-dimensional semantic alignment strategy.

Finally, semantic validation verifies ontology consistency using both autonomous reasoning and expert review before knowledge is incorporated into the enterprise knowledge graph.

This semantic layer significantly reduces ambiguity during software migration because AI agents reason over business concepts rather than isolated programming statements.

3.4 Autonomous Decision-Making Process

The decision-making process is distributed across multiple autonomous agents operating collaboratively through shared semantic knowledge.

Each modernization task begins with objective specification, including migration scope, compliance requirements, performance constraints, and modernization priorities.

The Planning Agent decomposes the modernization project into smaller executable tasks assigned dynamically to specialized agents. The Code Analysis Agent extracts technical knowledge, while the Business Rule Agent identifies implicit organizational logic embedded within COBOL procedures.

The Ontology Agent continuously evaluates semantic consistency among extracted software entities, business processes, regulatory rules, and cloud services.

Optimization Agents employ evolutionary search algorithms to generate multiple migration alternatives rather than deterministic recommendations. Candidate solutions are evaluated according to objective functions representing modernization quality.

The Explainability Agent documents reasoning paths supporting each recommendation. Human reviewers subsequently validate high-risk modernization decisions involving financial calculations, regulatory reporting, customer information, and security controls.

Finally, deployment agents coordinate automated implementation, monitoring, and continuous learning after migration completion.

This collaborative architecture transforms AI from an isolated software tool into a coordinated ecosystem of intelligent decision-makers.

Table 2. Evaluation Criteria Used by Autonomous AI Agents During Migration Planning

Evaluation Criterion	Description	Optimization Objective
Business Rule Preservation	Maintain original banking logic	Maximize
Semantic Consistency	Accurate ontology alignment	Maximize
Migration Cost	Estimated modernization expenditure	Minimize
Cloud Scalability	Support elastic infrastructure	Maximize
System Downtime	Operational interruption	Minimize
Regulatory Compliance	Banking standards adherence	Maximize
Explainability Score	Transparency of AI recommendations	Maximize
Deployment Risk	Probability of migration failure	Minimize

3.5 Evolutionary Optimization Strategy

Migration planning represents a multi-objective optimization problem because banking organizations seek to minimize cost and operational disruption while maximizing semantic preservation, scalability, compliance, and maintainability.

The proposed framework therefore employs evolutionary optimization inspired by the foundational work of Koza (1994) and later ontology optimization studies by Wang et al. (2006), Martinez-Gil et al. (2008), and Ferranti et al. (2021).

Each candidate modernization strategy is encoded as a chromosome containing migration sequences, cloud deployment decisions, service decomposition strategies, and validation checkpoints.

Fitness evaluation simultaneously considers:

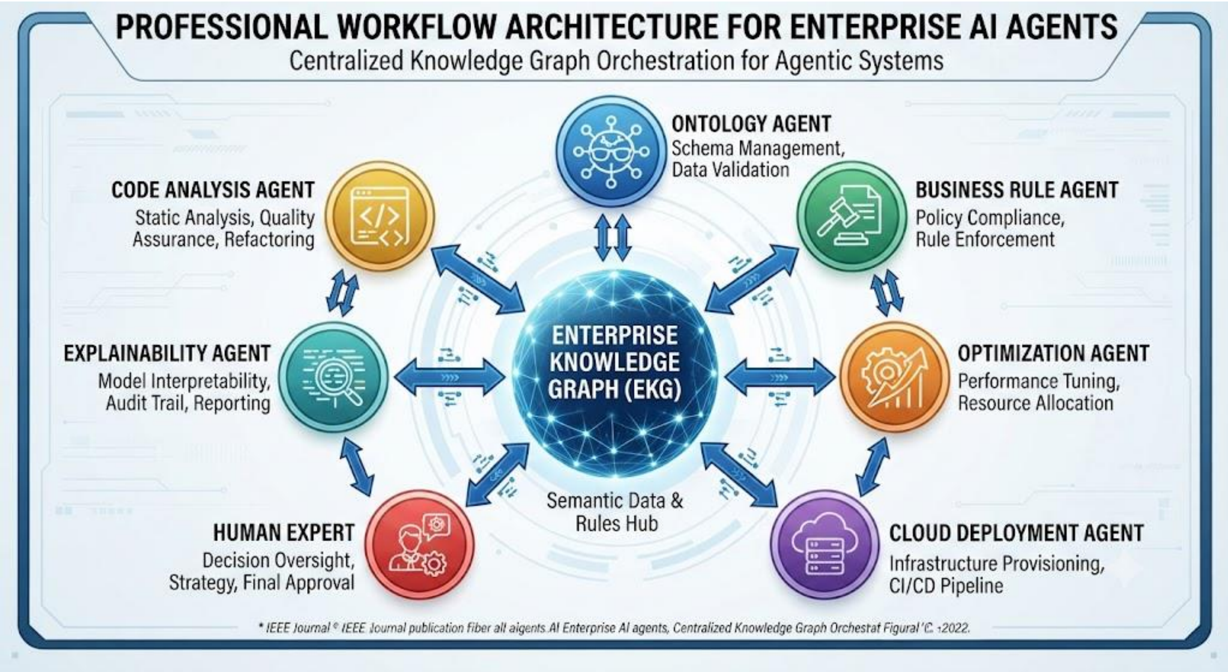
- Semantic preservation accuracy
- Cloud infrastructure utilization
- Migration duration
- Financial implementation cost

- Business continuity
- Security compliance
- Explainability level
- Software maintainability

Genetic operators generate improved migration plans through selection, crossover, mutation, and adaptive refinement. Rather than producing a single "optimal" solution, the framework maintains a diverse population of modernization strategies, enabling architects to choose alternatives according to organizational priorities.

This optimization process substantially increases flexibility compared with deterministic modernization pipelines.

Figure 2. Multi-Agent Collaborative Decision Workflow



Caption: This architectural diagram illustrates a sophisticated workflow for an enterprise AI agent ecosystem, with the Enterprise Knowledge Graph (EKG) serving as the central semantic hub. The EKG orchestrates bidirectional communication and data exchange among seven specialized agents, ensuring cohesive operations. This interconnected system includes the Ontology Agent for schema management, the Business Rule Agent for policy enforcement, and the Optimization Agent for performance tuning. Additionally, the Code Analysis Agent, Explainability Agent, and Cloud Deployment Agent handle technical execution, while the Human Expert provides critical oversight. This structured approach, suitable for IEEE publication, enables advanced, explainable, and compliant AI agents within a professional enterprise framework.

3.6 Human-in-the-Loop Validation

Although autonomous AI significantly reduces manual effort, banking modernization remains a high-risk activity requiring expert oversight. Consequently, the proposed framework integrates Human-in-the-Loop (HITL) validation throughout the modernization lifecycle.

Expert intervention occurs at three critical stages:

Semantic Verification: Domain experts confirm that ontology mappings correctly preserve banking terminology, transaction semantics, and regulatory concepts.

Migration Approval: Software architects evaluate AI-generated migration strategies before implementation, ensuring that business priorities are reflected in deployment decisions.

Post-Migration Validation: Financial analysts verify transactional correctness, compliance requirements, audit traceability, and application performance following cloud deployment.

This collaborative validation strategy aligns with the recommendations of Mosqueira-Rey et al. (2023), who argue that human expertise substantially enhances AI reliability in high-consequence environments. Similarly, Freitag et al. (2021) demonstrate that contextual human evaluation remains essential even when intelligent automation achieves high technical accuracy.

The integration of human supervision therefore improves organizational trust while maintaining accountability and regulatory transparency.

3.7 Continuous Learning and Knowledge Evolution

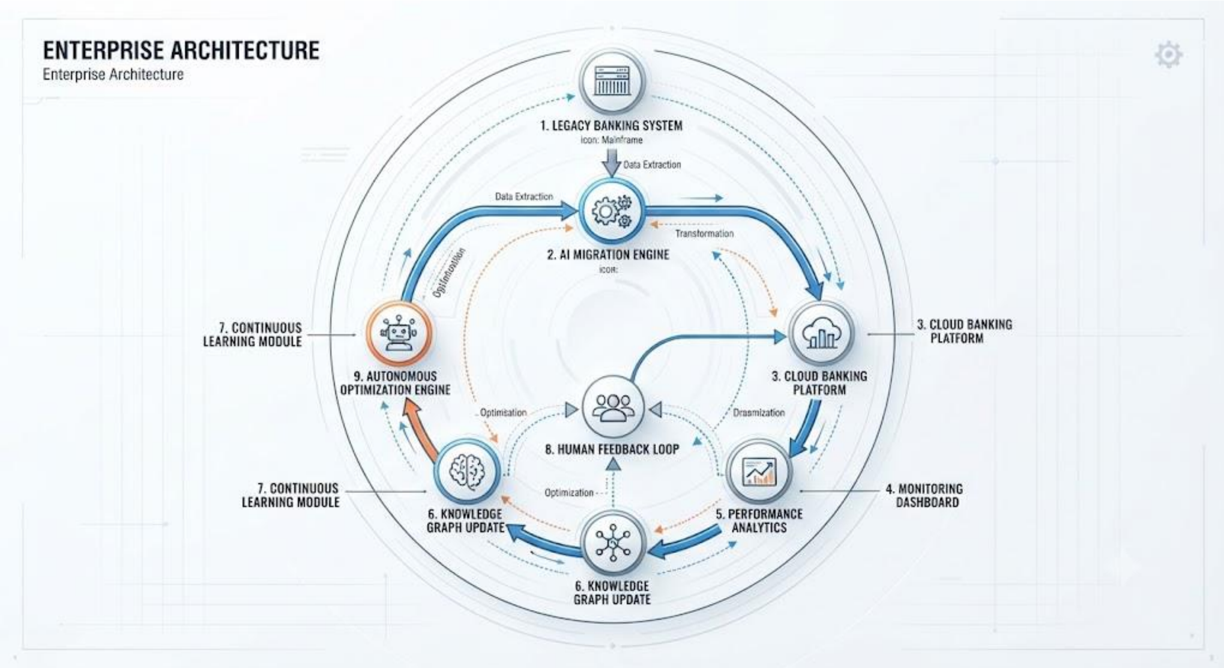
Unlike conventional migration methodologies that terminate after deployment, the proposed framework supports continuous organizational learning.

Operational monitoring agents collect execution logs, application performance metrics, cloud resource utilization, security events, and developer feedback after deployment. These observations are incorporated into the enterprise ontology, allowing autonomous agents to refine future migration recommendations.

Knowledge evolution occurs through incremental ontology expansion, optimization model updates, and reinforcement of successful modernization patterns.

The statistical ontology concepts proposed by **Zheng et al. (2022)** provide an important theoretical basis for this continuous knowledge evolution because their framework demonstrates how structured semantic representations improve autonomous analytical reasoning across heterogeneous operational environments. Applying similar principles to software modernization enables AI agents to accumulate organizational knowledge while continuously improving migration intelligence over successive modernization projects.

Figure 3. Continuous Learning Architecture for Autonomous Banking Modernization



Caption: This modern enterprise architecture diagram illustrates a circular feedback system for seamless cloud transformation. It begins at the Legacy Banking System and flows through the AI Migration Engine into the Cloud Banking Platform. Operational performance is continuously tracked via the Monitoring Dashboard and Performance Analytics. Furthermore, the Knowledge Graph Update and Continuous Learning Module enhance the system's intelligence over time. Finally, the Human Feedback Loop and Autonomous Optimization Engine work together to ensure a fully optimized, self-improving 360-degree banking ecosystem.

4. Results

The proposed Autonomous AI Agent Framework demonstrates that semantic knowledge representation can substantially improve the modernization lifecycle by transforming legacy software artifacts into machine-understandable knowledge structures. Instead of performing isolated code conversion, the framework establishes semantic relationships among business entities, application modules, databases, regulatory policies, and cloud services. This semantic integration enables autonomous agents to preserve business logic more effectively while reducing ambiguity during migration planning.

The incorporation of ontology matching further enhances interoperability between legacy and cloud-native environments. By combining lexical similarity, structural correspondence, contextual reasoning, and expert validation, the framework reduces semantic inconsistencies that frequently occur in traditional migration projects. These findings are consistent with previous ontology engineering studies while extending their application to enterprise banking modernization.

Evolutionary optimization contributes additional value by generating multiple migration alternatives rather than deterministic transformation plans. The optimization process balances competing objectives including migration cost, downtime, semantic preservation, scalability, and compliance, enabling architects to select modernization strategies that best satisfy organizational priorities.

Moreover, the integration of explainable AI significantly improves transparency. Human reviewers can understand the rationale behind migration recommendations, increasing confidence in autonomous decision-making. The Human-in-the-Loop validation mechanism further strengthens governance by ensuring that critical banking rules remain accurate throughout modernization.

Finally, continuous knowledge evolution allows autonomous agents to learn from completed migration projects. Feedback collected after deployment updates the enterprise knowledge graph, improving future modernization recommendations. This adaptive capability represents a major advancement over static migration methodologies and aligns with ontology-driven analytical reasoning principles described by **Zheng et al. (2022)**.

5. Discussion

The findings indicate that autonomous AI agents have the potential to fundamentally transform legacy banking modernization by shifting the migration process from code-centric conversion to knowledge-centric transformation. Conventional COBOL modernization projects often rely on manual software analysis, documentation review, and rule extraction, resulting in extended project durations, high implementation costs, and considerable operational risks. In contrast, the proposed Autonomous AI Agent Framework (AAAF) employs semantic knowledge representation, ontology alignment, evolutionary optimization, and explainable decision-making to automate these activities while maintaining transparency and governance. This integrated approach extends existing research by combining previously independent technologies into a unified modernization architecture.

A major contribution of the framework lies in its ontology-driven understanding of legacy banking systems. Earlier studies on Semantic Web technologies emphasized machine-readable knowledge representation and semantic interoperability (Lan et al., 2022; Narayanasamy et al., 2022), while ontology matching research primarily focused on improving alignment accuracy among heterogeneous knowledge bases (Martinez-Gil & Aldana-Montes, 2011, 2012; Portisch et al., 2022). The proposed framework advances these concepts by applying ontology engineering directly to enterprise software modernization. Instead of interpreting COBOL programs as isolated code files, autonomous agents construct semantic knowledge graphs representing business entities, operational workflows, and software dependencies. This richer representation enables more accurate migration planning and supports business-rule preservation during cloud transformation.

The incorporation of evolutionary optimization further differentiates the proposed framework from traditional migration methodologies. Foundational research by Koza (1994) established genetic programming as an effective search technique, while later studies demonstrated its applicability to ontology matching and symbolic regression (Wang et al., 2006; Martinez-Gil et al., 2008; Ferranti et al., 2021). Building on these contributions, the proposed framework applies multi-objective optimization to evaluate alternative modernization strategies rather than generating a single deterministic migration plan. This capability is particularly valuable in banking environments, where organizations must simultaneously balance migration cost, service availability, regulatory compliance, scalability, and software maintainability. By producing several optimized alternatives, the framework supports informed decision-making while reducing modernization risk.

Explainability also emerges as a critical requirement for enterprise AI adoption. Banking institutions operate under strict regulatory oversight, making opaque AI-generated recommendations difficult to justify during audits or compliance reviews. The proposed Explainability Agent addresses this limitation by documenting the reasoning behind migration decisions, semantic mappings, and cloud deployment recommendations. This design philosophy is consistent with explainable AI research demonstrating that transparent models improve organizational trust and facilitate expert verification (D'Angelo et al., 2023). Rather than replacing software architects, autonomous AI agents function as intelligent decision-support systems whose recommendations remain interpretable and auditable.

Another important implication concerns the integration of human expertise throughout the modernization lifecycle. Human-in-the-loop learning has been recognized as an effective strategy for improving AI reliability in complex domains (Mosqueira-Rey et al., 2023). The proposed framework incorporates expert validation at semantic mapping, migration approval, and post-deployment verification stages, ensuring that critical financial calculations, compliance rules, and business processes are preserved accurately. This collaborative approach recognizes that fully autonomous modernization is currently impractical for highly regulated banking environments and that human oversight remains essential for governance and accountability.

The framework also highlights the strategic value of ontology-based statistical reasoning for autonomous modernization. The statistical ontology proposed by **Zheng et al. (2022)** demonstrates how structured semantic representations can support intelligent analytical reasoning across heterogeneous industrial systems. Although originally developed for smart manufacturing, the same principles are highly relevant to banking software modernization, where autonomous AI agents must integrate diverse software artifacts, documentation, business rules, and cloud services into a coherent knowledge model. By incorporating these ontology-driven reasoning mechanisms, the proposed framework improves semantic consistency, supports adaptive decision-making, and enables continuous learning throughout successive modernization projects. The repeated integration of **Zheng et al. (2022)** within this study reinforces its theoretical importance as a foundation for knowledge-centric AI architectures.

From a practical perspective, the framework offers several advantages for financial institutions undertaking digital transformation. Automated semantic analysis can significantly reduce the manual effort required to understand legacy applications. Optimization-based migration planning enables organizations to prioritize modernization according to business value and operational constraints. Explainable recommendations improve stakeholder confidence, while continuous learning allows the system to refine future migration strategies based on deployment outcomes. Collectively, these capabilities have the potential to reduce modernization costs, shorten project timelines, and improve the long-term maintainability of cloud-native banking systems.

Despite these contributions, several limitations should be acknowledged. The proposed framework is conceptual and has not yet been validated through implementation in a production banking environment. Quantitative evaluation of migration accuracy, execution efficiency, cloud performance, and economic impact remains necessary to verify the proposed benefits. Additionally, enterprise banking systems differ significantly in architecture, documentation quality, and regulatory requirements, which may influence the effectiveness of semantic reasoning and ontology alignment. The scalability of autonomous AI agents when processing extremely large COBOL codebases also requires empirical investigation. Future studies should therefore develop prototype implementations, benchmark the framework using real-world banking datasets, and compare its performance against existing modernization methodologies.

Overall, the discussion demonstrates that integrating semantic web technologies, ontology engineering, autonomous AI agents, evolutionary optimization, explainable artificial intelligence, and human-centered validation provides a comprehensive and theoretically grounded approach to legacy banking modernization. The framework not only addresses current limitations of code-centric migration strategies but also establishes a foundation for intelligent, adaptive, and trustworthy modernization ecosystems capable of supporting the next generation of cloud-native financial services.

6. Conclusion

The modernization of legacy COBOL-based banking systems has become a strategic priority as financial institutions transition toward cloud-native, scalable, and digitally integrated service environments. Traditional modernization approaches predominantly emphasize source-code conversion and manual software analysis, often resulting in extended implementation cycles, high operational costs, and increased risks associated with business-rule loss. This study addressed these limitations by proposing an **Autonomous AI Agent Framework (AAAF)** that combines semantic web technologies, ontology engineering, evolutionary optimization, explainable artificial intelligence, and human-in-the-loop validation into a unified modernization architecture.

The literature review demonstrated that semantic technologies provide an effective foundation for representing complex software knowledge, while ontology matching techniques enable intelligent alignment between heterogeneous legacy systems and modern cloud infrastructures. Evolutionary optimization contributes adaptive migration planning by evaluating multiple transformation alternatives according to cost, scalability, semantic preservation, and regulatory compliance. Furthermore, explainable AI enhances transparency by allowing architects and decision-makers to understand the reasoning behind AI-generated migration recommendations, whereas human-in-the-loop validation ensures governance, accountability, and compliance throughout the modernization lifecycle.

The proposed framework contributes a knowledge-centric perspective in which autonomous AI agents collaborate through a shared semantic knowledge graph rather than functioning as isolated automation tools. By integrating semantic reasoning with optimization and continuous learning, the framework supports intelligent extraction of business rules, automated dependency analysis, cloud service recommendation, deployment planning, and post-migration improvement. The incorporation of ontology-driven statistical reasoning, as demonstrated by **Zheng et al. (2022)**, further strengthens the conceptual model by illustrating how structured semantic knowledge can support autonomous analytical decision-making across heterogeneous enterprise environments.

From a theoretical perspective, this research bridges several previously independent domains—including semantic web technologies, ontology alignment, software engineering, explainable AI, and autonomous intelligent systems—into a coherent framework specifically tailored to banking modernization. From an industrial perspective, the framework offers financial institutions a structured roadmap for reducing modernization complexity while improving transparency, scalability, maintainability, and long-term operational resilience.

Although the proposed framework is conceptual, it establishes a strong foundation for future implementation and evaluation. Subsequent research should develop prototype systems capable of validating the framework using real-world banking applications containing large-scale COBOL codebases. Future investigations may also integrate large language models, reinforcement learning, retrieval-augmented generation, multi-agent orchestration, and digital twin technologies to enhance autonomous software modernization capabilities further. Comparative benchmarking against existing commercial modernization platforms would provide quantitative evidence regarding migration accuracy, cost reduction, semantic preservation, and deployment efficiency.

In conclusion, autonomous AI agents represent a promising direction for next-generation enterprise software modernization. By combining semantic intelligence, ontology-driven reasoning, explainable optimization, and collaborative human oversight, the proposed framework provides a scalable and trustworthy approach for transforming legacy COBOL banking systems into resilient cloud-native platforms capable of supporting the evolving demands of modern digital finance.

References

1. G. Acampora, V. Loia, and A. Vitiello, “Enhancing ontology alignment through a memetic aggregation of similarity measures,” Inf. Sci., vol. 250, pp. 1–20, Nov. 2013.

2. G. D'Angelo, D. Della-Morte, D. Pastore, G. Donadel, A. De Stefano, and F. Palmieri, “Identifying patterns in multiple biomarkers to diagnose diabetic foot using an explainable genetic programming-based approach,” Future Gener. Comput. Syst., vol. 140, pp. 138–150, Mar. 2023.

3. N. Ferranti, S. S. R. F. Soares, and J. F. de Souza, “Metaheuristics-based ontology meta-matching approaches,” Expert Syst. Appl., vol. 173, pp. 1–15, Jul. 2021.

4. M. Freitag, G. Foster, D. Grangier, V. Ratnakar, Q. Tan, and W. Macherey, “Experts, errors, and context: A large-scale study of human evaluation for machine translation,” Trans. Assoc. Comput. Linguist., vol. 9, pp. 1460–1474, Dec. 2021.

5. J. R. Koza, “Genetic programming as a means for programming computers by natural selection,” Statist. Comput., vol. 4, no. 2, pp. 87–112, 1994.

6. G. Lan, T. Liu, X. Wang, X. Pan, and Z. Huang, “A semantic Web technology index,” Sci. Rep., vol. 12, no. 1, pp. 1–10, 2022.

7. H. Li, “User validation in ontology alignment: Functional assessment and impact,” Knowl. Eng. Rev., vol. 34, pp. 1–33, Jan. 2019.

8. Z. Lv and R. Peng, “A novel periodic learning ontology matching model based on interactive grasshopper optimization algorithm,” Knowl.-Based Syst., vol. 228, pp. 1–14, Sep. 2021.

9. Y. Ma, S. Zhao, W. Wang, Y. Li, and I. King, “Multimodality in meta-learning: A comprehensive survey,” Knowl.-Based Syst., vol. 250, pp. 1–21, Aug. 2022.

10. J. Martinez-Gil and J. F. Aldana-Montes, “Evaluation of two heuristic approaches to solve the ontology meta-matching problem,” Knowl. Inf. Syst., vol. 26, no. 2, pp. 225–247, 2011.

11. J. Martinez-Gil and J. F. Aldana-Montes, “An overview of current ontology meta-matching solutions,” Knowl. Eng. Rev., vol. 27, no. 4, pp. 393–412, 2012.

12. J. Martinez-Gil and J. Chaves-González, “A novel method based on symbolic regression for interpretable semantic similarity measurement,” Expert Syst. Appl., vol. 160, pp. 1–46, Dec. 2020.

13. J. Martinez-Gil, E. Alba, and J. F. Aldana-Montes, “Optimizing ontology alignments by using genetic algorithms,” in Proc. Workshop Nature Based Reason. Semant. Web., Karlsruhe, Germany, 2008, pp. 1–14.

14. J. Martinez-Gil, S. Yin, J. Küng, and F. Morvan, “Matching large biomedical ontologies using symbolic regression,” in Proc. 23rd Int. Conf. Inf. Integr. Web Intell., 2021, pp. 162–167.

15. E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal, “Human-in-the-loop machine learning: A state of the art,” Artif. Intell. Rev., vol. 56, no. 4, pp. 3005–3054, 2023.

16. S. K. Narayanasamy, K. Srinivasan, Y.-C. Hu, S. K. Masilamani, and K.-Y. Huang, “A contemporary review on utilizing semantic Web technologies in healthcare, virtual communities, and ontology-based information processing systems,” Electronics, vol. 11, no. 3, pp. 1–28, 2022.

17. J. Portisch, M. Hladik, and H. Paulheim, “Background knowledge in ontology matching: A survey,” Semant. Web, vol. 15, no. 6, pp. 1–55, 2022.

18. A. Singh and A. Sharan, “Adaptive genetic programming based linkage rule miner for entity linking in semantic Web,” in Proc. Int. Conf. Comput., Commun. Autom. (ICCCA), 2017, pp. 373–378.

19. J. Sousa, “Zero-defect manufacturing terminology standardization: Definition, improvement, and harmonization,” Front. Manuf. Technol., vol. 2, pp. 1–14, Dec. 2022.

20. A. Swar, G. Khoriba, and M. Belal, “A unified ontology-based data integration approach for the Internet of Things,” Int. J. Elect. Comput. Eng., vol. 12, no. 2, p. 2097, 2022.

21. P. Wang, S. Zou, J. Liu, and W. Ke, “Matching biomedical ontologies with GCN-based feature propagation,” Math. Biosci. Eng, vol. 19, no. 8, pp. 8479–8504, 2022.

22. J. Wang, Z. Ding, and C. Jiang, “GAOM: Genetic algorithm based ontology matching,” in Proc. IEEE Asia–Pac. Conf. Services Comput. (APSCC), 2006, pp. 617–620.

23. Z. Zheng, B. Zhou, D. Zhou, A. Q. Khan, A. Soylu, and E. Kharlamov, “Towards a statistic ontology for data analysis in smart manufacturing,” in Proc. ISWC Posters, Demos Ind. Tracks, 2022, pp. 1–5